Dossier PLSQL

set serveroutput on;

--Question n°1--------

-- Créer une procédure pour inscrire un adhérent à une activité. Ajouter le nombre d'invités conviés.

--L'inscription sera refusée si la date limite d'inscription est dépassée, ou s'il y a plus de 50

--participants à l'activité

create or replace procedure inscrire\_adh\_activite

(

v\_id\_adh in membre.id\_membre%type,

v\_id\_act in activite.id\_activite%type,

v\_nb\_inv in participer.nombre\_invites%type

)

as

v\_date\_activite date;

v\_date\_du\_jour date;

date\_limite\_inscription date;

v\_nombre\_de\_participants number(10):=0;

begin

select date\_activite into v\_date\_activite from activite where id\_activite = v\_id\_act;

SELECT SYSDATE INTO v\_date\_du\_jour FROM dual;

select count(id\_membre)+sum(nombre\_invites) into v\_nombre\_de\_participants

from participer

where id\_activite = v\_id\_act;

date\_limite\_inscription := v\_date\_activite -5;

dbms\_output.put\_line('v\_date\_activite '|| v\_date\_activite);

dbms\_output.put\_line('date\_limite\_inscription '|| date\_limite\_inscription);

dbms\_output.put\_line('v\_nombre\_de\_participants ' || v\_nombre\_de\_participants);

if((v\_date\_du\_jour > date\_limite\_inscription)) then

dbms\_output.put\_line('la date de l''activite est trop rapprochée, ou l''activité est déjà passée');

RAISE\_APPLICATION\_ERROR (-20001, 'Date de l''activité trop rapprochée ou dépassée');

end if;

if (v\_nombre\_de\_participants>50) then

dbms\_output.put\_line('Trop de monde dans l''activité');

RAISE\_APPLICATION\_ERROR(-20002, 'le nombre de participants est supérieur à 50');

end if;

--dbms\_output.put\_line('On va inscrire l''adhérent à l''activité');

--dbms\_output.put\_line('Mais on va d''abord tester qu''il n''est pas déjà inscrit à cette même activité');

if(verifinscription(v\_id\_adh,v\_id\_act)) then

--dbms\_output.put\_line('l''adhérent n''est pas déjà inscrit à cette activité, on va donc l''y inscrire');

insert into participer (id\_membre, id\_activite, date\_inscription, nombre\_invites) values (v\_id\_adh,v\_id\_act,v\_date\_du\_jour, v\_nb\_inv);

--mais il faudrait également connaître le nombre d'invités et leurs noms, que je vais ignorer ici.

else

RAISE\_APPLICATION\_ERROR(-20002, 'l''adhérent est déjà inscrit');

end if;

end inscrire\_adh\_activite;

---------------------

-- Vérifier qu'un adhérent n'est pas déjà inscrit à une activité

create or replace function verifinscription

(

v\_id\_adh in membre.id\_membre%type,

v\_id\_act in activite.id\_activite%type

)

return boolean

as

v\_adherent\_estinscrit number(1);

begin

--select count(\*) into v\_adherent\_existe from membre where id\_membre = v\_id\_adh;

select count(\*) into v\_adherent\_estinscrit from participer where id\_membre = v\_id\_adh and id\_activite = v\_id\_act;

if(v\_adherent\_estinscrit =0)then

return true;

else return false;

end if;

end verifinscription;

-----lancer l'inscription avec pour paramètres l'id adhérent, l'id de l'activité et le nombre de

-----participants

begin

inscrire\_adh\_activite(3,8,3);

end;

--Résultat---------------------------------------------------------------------------------------

v\_date\_activite 01/10/23

date\_limite\_inscription 26/09/23

v\_nombre\_de\_participants 5

Procédure PL/SQL terminée.

--Question n°2----------------

--Créer une fonction pour valider une adresse e-mail

--Fonction permettant de vérifier qu'une adresse mail est valide

CREATE or replace FUNCTION verif\_mail (email varchar2) RETURN VARCHAR2

IS

ESTVALIDE BOOLEAN;

BEGIN

ESTVALIDE := REGEXP\_LIKE (email,'^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$');

IF ESTVALIDE

THEN

return 'VALIDE';

ELSE

return 'NON VALIDE';

END IF;

END;

begin

dbms\_output.put\_line(verif\_mail('emilie.paniagua@gmail.com'));

end;

--Résultat-------

VALIDE

Procédure PL/SQL terminée.

-------Question 3------------

/\* Créer une procédure ou une fonction qui renvoie la liste des adhérents sous forme d'une chaine de caractères type fichier CSV à raison d'une ligne par adhérent sous la forme "nom; prénom; mail; tel"\*/

create or replace function extraire\_membres\_csv

return VARCHAR2

as

v\_count\_membres\_total number(10);

v\_nom membre.nom%type;

v\_prenom membre.prenom%type;

v\_email membre.email%type;

v\_tel membre.NoTel%type;

v\_lines varchar2(5000);

begin

--cette procédure extrait les valeurs des membres de tous les adhérents de la table membre pour le restituer au format "nom; prénom; mail; tel" \n

select count(\*) into v\_count\_membres\_total from membre;

for I in 1..v\_count\_membres\_total loop

--dbms\_output.put\_line('I : '|| i);

select nom, prenom, email, notel into v\_nom, v\_prenom, v\_email, v\_tel from membre where id\_membre=I;

v\_lines := v\_lines ||'"' ||v\_nom||';'||v\_prenom||';'||v\_email||';'||v\_tel||'"'||chr(10);

end loop;

return v\_lines;

end;

begin

dbms\_output.put\_line(extraire\_membres\_csv);

end;

--Résultat : --

"Duchampt;Jean-luc;jl@duchampt.com;(502) 873-2752""Ergot;Françoise;ergot.f@free.fr;(261) 915-7516""Dubois;Alain;Alain.Dubois9988@gmail.com;(622) 294-6408""Dupont;Amandine;Amandine.Dupont@mail.ru;(990) 777-6908""Arnaud;Rémi;Remi.Arnaud9966@yahoo.fr;(406) 824-5110""Marquet;Françoise;Francoisedsmarquet@laposte.fr;(501) 263-5617""Antonini;Florence;Florence.Antonini@gmx.de;(620) 298-0312""Leroy;Jean-François;LeroyJF@yahoo.fr;(230) 162-6732""Desmartins;Jean-Luc;DesmartinsJL@gmail.com;(343) 732-7554""Desjardins;Sylvie;Desjardins.Sylvie01@orange.fr;(185) 716-1788""Ayrald;Etienne;Ayrald5601Etienne@yahoo.fr;(278) 890-7589""Milham;Ed;;(808) 667-1170""Tate;Jessica;;(890) 213-6063"